UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/815,897	03/31/2004	Xuebin Yao	P19003	9318

46915 7590 09/03/2008

KONRAD RAYNES & VICTOR, LLP.
ATTN: INT77
315 SOUTH BEVERLY DRIVE, SUITE 210
BEVERLY HILLS, CA 90212

EXAMINER	
ZHANG, SHIRLEY X	
ART UNIT	**PAPER NUMBER**
2144	
MAIL DATE	**DELIVERY MODE**
09/03/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/815,897	YAO ET AL.	
	Examiner	Art Unit	
	SHIRLEY X. ZHANG	2144	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 May 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,8-14,17-26 and 29-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,8-14,17-26 and 29-34* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This final office action is prepared in response to the applicant's amendments and arguments filed on May 21, 2008 as a reply to the non-final office action mailed on February 21, 2008.

Claims 6-7, 15-16 and 27-28 have been cancelled;

Claims 31-34 are newly added.

Claims 1, 10, 19, 22-26, and 29-30 have been amended;

Claims 1-5, 8-14, 17-26, 29-34 are now pending;

Response to Arguments

Applicant's arguments and amendments filed on May 21, 2008 have been carefully considered but are deemed moot in view of the new grounds of rejection as explained here below, which is necessitated by Applicant's substantial amendments to the claims that significantly affected the scope thereof, and will require further search and consideration.

Accordingly, THIS ACTION IS MADE FINAL. See MPEP 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

1. **The 101 rejections of claims 22-30 are maintained**. Paragraph [0060] of Applicant's specification disclosed "transmission medium" including network transmission lines, wireless transmission media and signals propagating through space, radio waves and infrared signals, etc., which indicates that Applicant intends the computer readable storage medium to include

transmission medium and signals that are considered unpatentable subject matter under 35 U.S.C. 101.

2. Applicant's remarks regarding the newly amended claims 1, 10, 19 and 22 are deemed moot in view of the new ground of rejections of these claims under 35 U.S.C. 103(a) as presented below.

3. The dependent claims are also rejected under 35 U.S.C. 103(a) with Examiner's rationale presented in the section "Claim Rejections – 35 USC 103" below.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. **Claims 22-30** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 22 recite(s) the limitation, "a computer readable storage medium."

Paragraph [0060] of Applicant's specification disclosed "transmission medium" including network transmission lines, wireless transmission media and signals propagating through space, radio waves and infrared signals, etc., which provides evidence that Applicant intends the computer readable storage medium to include transmission medium and signals that are considered unpatentable subject matter under 35 U.S.C. 101.

Claims 23-30 are dependent on claim 22, but fail to further limit claim 22 to statutory subject matter, therefore inherit the 35 U.S.C. 101 issue of the independent claim.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. **Claims 1-4, 8-13, 17-25 and 29-30** are rejected under 35 U.S.C. 103(a) as obvious over U.S. patent application publication no. 2003/0158906 to **Hayes**, in view of Yeh et al. (white paper "Introduction to TCP/IP Offload Engine (TOE)" published in 2002, hereinafter "**Yeh**"), and Pinkerton et al. (US Patent No. 7,181,531, hereinafter "Pinkerton").

Regarding claim 1, Hayes discloses a method, comprising: requesting, by a network storage driver (Fig.10 and [0056] disclose a network application 172, which is equivalent to the network storage driver recited in the claim), a connection from an offload application ([0066] discloses that to establish a connection, in FIG. 7, network application 172 sends a request to a host resident offload task interface function 162. Here the host resident functions 162, 166 and 167 together is equivalent to the offload application recited in the claim because [0066] discloses that the AP and host resident TCP and +Application protocol processing functions 166, 158, 167, 159 are able to offload the network and application protocols that network application 172 uses), wherein the offload application interfaces with a first network stack implemented in an operating system (Fig. 10 discloses a first network stack that includes the components 118, 116 and 114 in a host operating system) and a second network stack implemented in a hardware device (Fig. 10

discloses a second protocol stack that includes the components 159, 156 and 170; [0056] further discloses that the host resident offload task interface function 162 communicates with host resident processing functions 164, 166 and 167 on one end and an AP resident offload task interface function 154 that controls the second network stack on the hardware on the other end);

receiving the connection from the offload application, wherein the received connection is an offloaded connection and is reserved for the network storage driver ([0066] discloses that in order to retrieve data from the network storage system, the network application, i.e. the network storage driver, must establish a connection, which is reserved for the network application; the last sentence of [0067] discloses that once the connection is established, a host resident task interface function 162 notifies a network application 172 of the connection); and

communicating data over the offloaded connection through the hardware device ([0068] discloses that after a connection has been established, network application 172 calls a host resident offload task interface function162 requesting that data be sent to network attached storage 16),

wherein the network storage driver is an iSCSI driver that implements the iSCSI protocol for communicating with a target storage device through the hardware device ([0038-0039] disclosed that the auxiliary processor offloads the transmission and reception of iSCSI data over the TCP/IP, and Fig. 7 disclosed that the network interface card is a regular NIC card without SCSI bus adapter, implying that the task interface implements the iSCSI protocol).

Hayes did not explicitly disclose:

wherein the first network stack and the second network stack do not implement an Internet Small Computer Systems Interface (iSCSI) protocol,

However, Yeh discloses in section "Introduction" on page 1 the approach of offloading TCP/IP protocol stack to the hardware while leaving application layer protocols such as iSCSI in the software on the host, as further disclosed in the section "Applications" on page 4.

It would have been obvious for one of ordinary skill in the art to combine Hayes and Yeh such that the first network stack and the second network stack do not implement an Internet Small Computer Systems Interface protocol. One would have been motivated to combine Hayes and Yeh so that the protocol stack on the hardware will be application-neutral and therefore has the flexibility of supporting various types of application layer protocols in the software on a need basis.

Hayes did not explicitly disclose

wherein the iSCSI driver comprises an iSCSI protocol layer and an iSCSI transport abstraction layer, wherein the iSCSI transport abstraction layer provides an abstracted transport interface such that the iSCSI protocol layer is not aware of any operating system and hardware transport specifics for communicating commands to the hardware device;

However, Pinkerton disclosed a method for offloading TCP/IP stack to a NIC card by using a chimney driver for direct communication between upper layer applications and TCP/IP offload engine hardware, and the upper edge of the chimney driver 312 is the NDIS API in Microsoft operating system. NDIS API is an abstracted, standard API that hides the underlying operating system and hardware transport specifics (Fig. 3 and column 9, lines 37-58)

One of ordinary skill in the art would have been motivated to combine Hayes and Pinkerton because both disclosed offloading network protocol stack to interface cards/devices (Hayes, "Abstract" and Pinkerton, "Abstract").

Therefore, it would have been obvious for one to incorporate Pinkerton's teaching of providing an abstracted standard device driver interface to the upper layer into Hayes such that Hayes' task interface consists of an iSCSI layer to interface with SCSI applications on the upper edge, and an abstracted layer on the lower edge to wrap and hide operating system and hardware specifics from the upper iSCSI layer. The benefit of doing so is that protocol stack and device driver can be modified and updated independently, making it easy to maintain the software, and also increasing the portability of the protocol stack and the device driver.

Regarding claim 2, the combination of Hayes, Yeh and Pinkerton disclosed the method of claim 1, wherein communicating the data over the offloaded connection further comprises: sending the data directly from the network storage driver to a hardware driver for the hardware device ([0068] discloses that to request that data be send to network attached storage, the network application calls a host resident offload task interface function, which then calls an auxiliary processor (AP) resident offload task interface with a service request so that the request can be processed by the protocol stack on the AP, bypassing the host protocol stack), wherein the network storage driver uses the second network stack implemented in the hardware device to communicate with a storage area network ([0038] discloses that the auxiliary processor offloads the reception of iSCSI data over the TCP/IP network protocol, performing all necessary TCP/IP

functions that occur during the normal course of a TCP/IP receive operation and all necessary iSCSI data movement functions).

Regarding claim 3, the combination of Hayes, Yeh and Pinkerton disclosed the method of claim 1, further comprising: releasing the offloaded connection to the offload application, wherein the offloaded connection is no longer reserved for the network storage driver ([0071] discloses that if the given TCP connection is to be reused and the error recovery has been completed, the TCP connection state can again be transferred from the host resident TCP protocol offload function 166 to the AP resident TCP protocol offload processing function 158, which implies that if the connection is not reused or the error recovery failed, the connection will be released).

Regarding claim 4, the combination of Hayes, Yeh and Pinkerton disclosed the method of claim 1 further comprising:

receiving the request for the connection at the offload application ([0066] further discloses that to establish a connection, in FIG. 7, network application 172 sends a request to a host resident offload task interface function 162 to open a TCP connection, where the host resident offload task interface function is equivalent to the offload application recited in the claim);

generating, by the offload application, the offloaded connection ([0067] discloses that to establish the connection, a host resident offload task interface function 162 calls a host resident TCP protocol offload processing function with a protocol service request),

reserving, by the offload application, the offloaded connection for the network storage driver ([0057] discloses that protocol state information is passed between host and the auxiliary processor, where [0103] further discloses that protocol state information is the computer data necessary to maintain a network connection by a protocol stack, i.e. it is the resources that must be reserved by the host resident TCP protocol offload processing function) and

sending the offloaded connection to the network storage driver ([0067] discloses that once a connection is established, a host resident task interface function 162 notifies a network application 172 of the connection).

Regarding claims 5, the combination of Hayes, Yeh and Pinkerton disclosed the method of claim 1, respectively, wherein the connection is a Transmission Control Protocol/Internet Protocol connection including state information describing the connection ([0066] discloses that a network application sends a request to open a TCP connection; [0067] further discloses that associated with each connection is state information describing the connection) sent from the offload application to the network storage driver ([0067] further disclose that once a connection is established, a host resident task interface function, i.e., the offload application, notifies a network application, i.e., the network storage driver) and wherein the state information includes a port address that is reserved for the network storage driver ([0106] discloses that a TCP connection is identified by the IP source address, destination address, source port and destination port).

Hayes did not explicitly disclose a file descriptor for the TCP connection.

However, Yeh discloses in section "Performance with TCP offload" on page 4 that TOE usually interface to the system above the transport layer with a socket interface in a sockets-based system, which implies that a socket handle, which is equivalent to a file descriptor, is created for each TCP connection.

It would have been obvious for one of ordinary skill to integrate Yeh's teaching of implementing a socket interface into Hayes so that each TCP connection in Hayes's invention includes a socket handle, or a file descriptor. One would have been motivated to combine Hayes and Yeh as such by the fact that both Hayes and Yeh taught about techniques of offloading TCP/IP stack onto hardware to expedite data processing for certain application layer protocols such as iSCSI, and socket interface is such a well-known technology in the art of TCP/IP networking that the combination would have yielded predictable results with reasonable expectation of success.

Regarding claim 8, the combination of Hayes, Yeh and Pinkerton disclosed the method of claim 1, wherein the first network stack and the second network stack comprise an Internet address family and a Transmission Control protocol implemented over an Internet Protocol network layer (Fig. 10), wherein the offload application can offload a network communication request to the second network stack in preference to the first network stack, and wherein a single stack behavior is maintained by the first and second network stacks to applications and network management utilities ([0069] discloses that the host resident offload task interface function 162 recognizes that this task is most efficiently accomplished by offloading it to an auxiliary

processor 152, and calls an AP resident offload task interface function 154 with a protocol service request).

Regarding claim 9, the combination of Hayes, Yeh and Pinkerton disclosed the method of claim 1, wherein the hardware device is a Transmission Control Protocol offload engine adapter ([0038] discloses that the auxiliary processor offloads the reception of iSCSI date over the TCP/IP network protocol and performs all necessary TCP/IP functions), and wherein a network communication request for communicating the data is processed faster in the second network stack in comparison to the first network stack ([0069] discloses that the host resident offload task interface function 162 recognizes that this task is most efficiently accomplished by offloading it to an auxiliary processor, which implies that it is processed faster in the second protocol stack in the auxiliary processor).

Claims 10-13 and 17-18 list substantially the same elements of **claims 1-4 and 8-9** but in system form rather than method form. Therefore, the supporting rationale of the rejection to claims 1-4 and 8-9 applies equally as well to claims 10-13 and 17-18.

Further regarding claim 10's limitation of "a system comprising a processor and program logic including code that is capable of causing the processor to be operable", Hayes disclosed in [0006] that the invention provides methods and apparatus for delivering selective offloading of protocol processing from a host computer to an offloading auxiliary processor and Fig. 7 discloses that the host computer comprises a CPU and memory. It is well known in the art that a

host computer as disclosed by Hayes inherently comprises program logic including code that is capable of causing the processor to be operable.

Claims 22-25 and 29-30 list substantially the same elements of **claims 1-4 and 8-9** but in the computer-readable form rather than method form. Therefore, the supporting rationale of the rejection to claims 1-4 and 8-9 applies equally as well to claims 22-25 and 29-30.

Furthermore, regarding claim 22's limitation of a computer readable storage medium having stored therein instructions capable of being executed by a machine, Hayes disclosed in [0006] that the invention provides methods and apparatus for delivering selective offloading of protocol processing from a host computer to an offloading auxiliary processor; Hayes further discloses in Fig. 7 that the host computer comprises a memory which inherently stores instructions capable of being executed by the CPU.

Claims 14 and 26 list all the same elements of **claim 5**, but in system or computer readable storage medium form rather than method form. Therefore, the supporting rationale of the rejection to **claim 5** applies equally as well to **claims 14 and 26.**

Regarding claim 19, Hayes discloses a system, comprising:

a computational platform (Fig. 4 discloses client computer 12 as a computational platform);

a storage controller implemented in the computational platform (Fig. 4 discloses the network attached storage device 16);

a processor coupled to the computational platform (Fig. 4 discloses a processor 28 on the client computer 12);

an offload adapter coupled to the computational platform (Fig. 8 discloses a network interface card that includes an auxiliary processor to offload protocol processing); and

program logic including code that is capable of causing the processor to be operable to:

request, by a network storage driver (Fig.10 and [0056] disclose a network application 172, which is equivalent to the network storage driver recited in the claim), a connection from an offload application ([0066] discloses that to establish a connection, in FIG. 7, network application 172 sends a request to a host resident offload task interface function 162. Here the host resident functions 162, 166 and 167 together is equivalent to the offload application recited in the claim because [0066] discloses that the AP and host resident TCP and +Application protocol processing functions 166, 158, 167, 159 are able to offload the network and application protocols that network application 172 uses), wherein the offload application interfaces with a first network stack implemented in an operating system (Fig. 10 discloses a first network stack that includes the components 118, 116 and 114 in a host operating system) and a second network stack implemented in the offload adapter (Fig. 10 discloses a second protocol stack that includes the components 159, 156 and 170; [0056] further discloses that the host resident offload task interface function 162 communicates with host resident processing functions 164, 166 and 167 on one end and an AP resident offload task interface function 154 that controls the second network stack on the hardware on the other end);

receive the connection from the offload application, wherein the received connection is an offloaded connection and is reserved for the network storage driver ([0066] discloses that in

order to retrieve data from the network storage system, the network application, i.e. the network storage driver, must establish a connection, which is reserved for the network application; the last sentence of [0067] discloses that once the connection is established, a host resident task interface function 162 notifies a network application 172 of the connection); and

communicate data over the offloaded connection through the offload adapter ([0068] discloses that after a connection has been established, network application 172 calls a host resident offload task interface function162 requesting that data be sent to network attached storage 16),

wherein the network storage driver is an iSCSI driver that implements the iSCSI protocol for communicating with a target storage device through the hardware device ([0038-0039] disclosed that the auxiliary processor offloads the transmission and reception of iSCSI data over the TCP/IP, and Fig. 7 disclosed that the network interface card is a regular NIC card without SCSI bus adapter, implying that the task interface implements the iSCSI protocol).

Hayes did not explicitly disclose:

wherein the first network stack and the second network stack do not implement an Internet Small Computer Systems Interface (iSCSI) protocol,

However, Yeh discloses in section "Introduction" on page 1 the approach of offloading TCP/IP protocol stack to the hardware while leaving application layer protocols such as iSCSI in the software on the host, as further disclosed in the section "Applications" on page 4.

It would have been obvious for one of ordinary skill in the art to combine Hayes and Yeh such that the first network stack and the second network stack do not implement an Internet

Small Computer Systems Interface protocol. One would have been motivated to combine Hayes and Yeh so that the protocol stack on the hardware will be application-neutral and therefore has the flexibility of supporting various types of application layer protocols in the software on a need basis.

Hayes did not explicitly disclose

wherein the iSCSI driver comprises an iSCSI protocol layer and an iSCSI transport abstraction layer, wherein the iSCSI transport abstraction layer provides an abstracted transport interface such that the iSCSI protocol layer is not aware of any operating system and hardware transport specifics for communicating commands to the hardware device;

However, Pinkerton disclosed a method for offloading TCP/IP stack to a NIC card by using a chimney driver for direct communication between upper layer applications and TCP/IP offload engine hardware, and the upper edge of the chimney driver 312 is the NDIS API in Microsoft operating system. It is know that NDIS API is an abstracted, standard API that hides the underlying operating system and hardware transport specifics (Fig. 3 and column 9, lines 37-58)

One of ordinary skill in the art would have been motivated to combine Hayes and Pinkerton because both disclosed offloading network protocol stack to interface cards/devices (Hayes, "Abstract" and Pinkerton, "Abstract").

Therefore, it would have been obvious for one to incorporate Pinkerton's teaching of providing an abstracted standard device driver interface to the upper layer into Hayes such that Hayes' task interface consists of an iSCSI layer to interface with SCSI applications on the upper

edge, and an abstracted layer on the lower edge to wrap and hide operating system and hardware specifics from the upper iSCSI layer. The benefit of doing such is that protocol stack and device driver can be modified and updated independently, making it easy to maintain the software, and also increasing the portability of the protocol stack and the device driver.

Regarding claim 20, the combination of Hayes, Yeh and Pinkerton disclosed the system of claim 19, wherein the program logic is further capable of causing the processor to be operable to: release the offloaded connection to the offload application, wherein the offloaded connection is no longer reserved for the network storage driver ([0071] discloses that if the given TCP connection is to be reused and the error recovery has been completed, the TCP connection state can again be transferred from the host resident TCP protocol offload function 166 to the AP resident TCP protocol offload processing function 158, which implies that if the connection is not reused or the error recovery failed, the connection will be released).

Regarding claim 21, the combination of Hayes, Yeh and Pinkerton disclosed the system of claim 19, wherein the program logic is further capable of causing the processor to be operable to:

receive the request for the connection at the offload application ([0066] further discloses that to establish a connection, in FIG. 7, network application 172 sends a request to a host resident offload task interface function 162 to open a TCP connection, where the host resident offload task interface function is equivalent to the offload application recited in the claim);

generate, by the offload application, the offloaded connection ([0067] discloses that to establish the connection, a host resident offload task interface function 162 calls a host resident TCP protocol offload processing function with a protocol service request);

reserve, by the offload application, the offloaded connection for the network storage driver ([0057] discloses that protocol state information is passed between host and the auxiliary processor, where [0103] further discloses that protocol state information is the computer data necessary to maintain a network connection by a protocol stack, i.e. it is the resources that must be reserved by the host resident TCP protocol offload processing function); and

send the offloaded connection to the network storage driver ([0067] discloses that once a connection is established, a host resident task interface function 162 notifies a network application 172 of the connection).

6. **Claims 31-34** are rejected under 35 U.S.C. 103(a) as obvious over Hayes, Yeh and Pinkerton as applied to claims 1, 10, 19 and 22 above, respectively, further in view of IETF Draft "iSCSI" published on January 10, 2003, hereinafter "**iSCSI-draft**".

Regarding claim 31, the combination of Hayes, Yeh and Pinkerton disclosed the method of claim 1, wherein transport interfaces included in the iSCSI transport abstraction layer are modified in response to a modification to the hardware device or the operating system, wherein no changes are made to the iSCSI protocol layer when changes are made to the iSCSI transport abstraction layer in response to the modification to the hardware device or the operating system (Yeh, "Introduction" and Pinkerton, Fig. 3 and column 9, lines 37-58, as addressed in the rejection of claim 1 above).

Hayes did not explicitly disclose:

wherein the iSCSI driver further comprises a Small Computer System Interface (SCSI) to iSCSI translation module that interfaces with an operating system SCSI stack and translates SCSI requests into iSCSI requests and then forwards the requests to the iSCSI protocol layer.

However, the above disclosure about iSCSI driver is what an iSCSI layer does by design, as disclosed in section 3 "Overview" of iSCSI-draft, the standard specification for iSCSI.

One of ordinary skill in the art would have been motivated to combine Hayes and iSCSI-draft because Hayes disclosed iSCSI as a possible embodiment of his invention.

It would have been obvious for one to implement Hayes' iSCSI according to the teaching of iSCSI-draft to make sure that the implementation is compliant with the standard specification and therefore will be interoperable with a third party iSCSI implementation, improving its market acceptance.

Claims 32-34 each list substantially the same elements of **claim 31,** but in system or computer readable storage medium form rather than method form. Therefore, the supporting rationale of the rejection to claim 31 applies equally as well to claims 32-34.

Conclusion

THIS ACTION IS FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SHIRLEY X. ZHANG whose telephone number is (571)270-5012. The examiner can normally be reached on Monday through Friday 7:30am - 5:00pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Vaughn can be reached on (571) 272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/S. X. Z./
Examiner, Art Unit 2144
8/28/2008
/William C. Vaughn, Jr./

Supervisory Patent Examiner, Art Unit 2144